Universidad ORT Uruguay
Facultad de Ingeniería

Fundamentos de Sistemas Ciberfísicos
Primer proyecto numérico

Tadeo Goday - 256686
Manuel Morandi - 271568
Matías Praderi - 269403

Tutores:
Carolina Allende
Martín Monteiro

2021

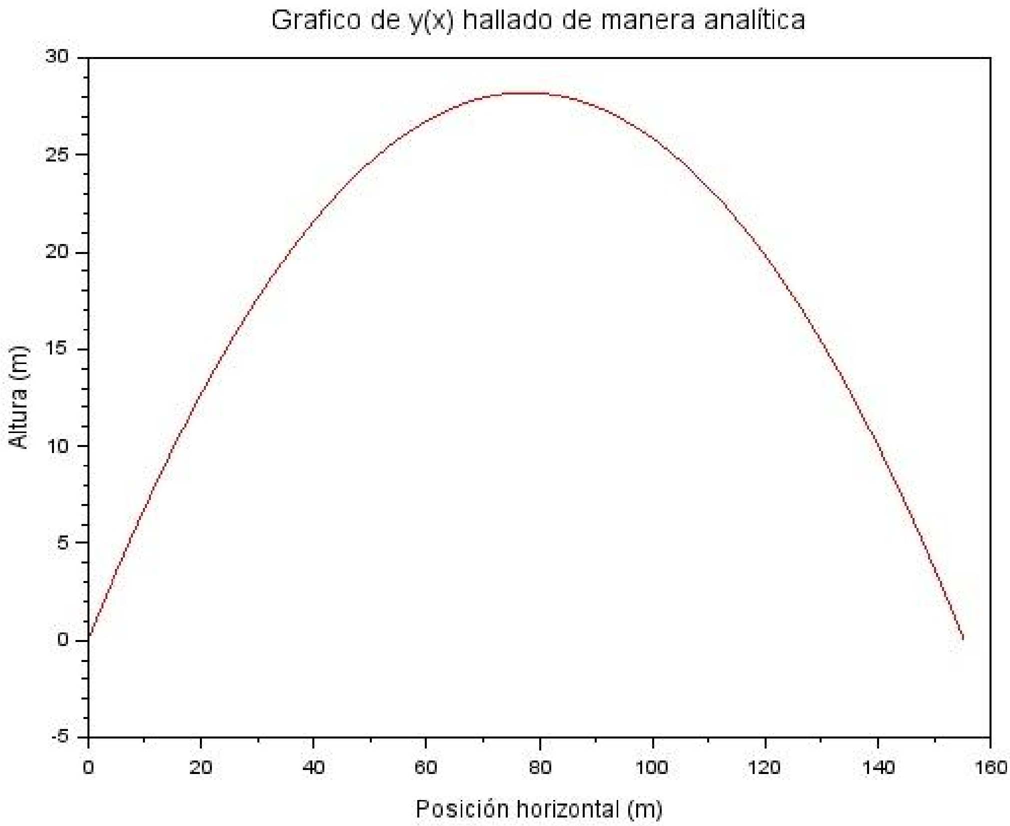
Nota: El código del programa se adjunta al final del informe.

1. Caso de proyectil sin influencia del aire:

a) Para esta parte, en la que se nos pide hallar la trayectoria de manera analítica, lo que haremos será utilizar los conceptos de cinemática dados en clase. Particularmente, usaremos la ecuación $y(x) = yo \ + voy(x - xo)/ - g(x - xo)^2/2vox^2$. Ingresamos los datos en Scilab y calculamos la trayectoria con ellos.

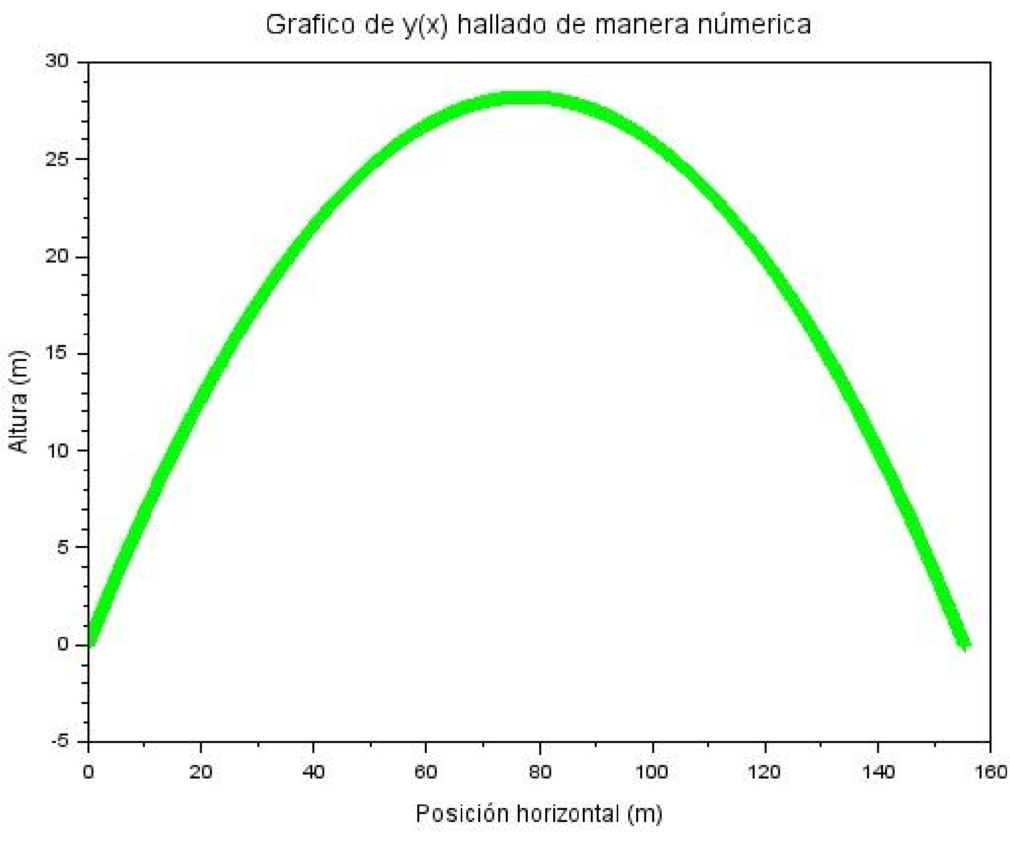
En el código, lo que tuvimos que hacer fue calcular x en función del tiempo e y en función del tiempo, y luego graficar lo hallado.

Gráfica resultante al hacerlo con $vo = 40 \frac{m}{s}$ y $\theta o = 36^o$



b) Se nos pide hallar la trayectoria utilizando el método de Euler. El método consiste en convertir una función diferencial en una recurrencia, usando la definición matemática de la derivada. Por ejemplo, sabemos que la derivada de la posición en y será la velocidad en y. Por la definición matemática $y' = vy = \lim_{\Delta t \to 0} \frac{y(t+\Delta t)-y(t)}{\Delta t}$. Conocemos el valor de vy, pero no y(t+Δt)-y(t). Si Δt es suficientemente pequeño podemos realizar algunas aproximaciones, llegando a la siguiente recurrencia: $vy * \Delta t + y(t) = y(t + \Delta t)$. Calculamos la posición en un momento dado a partir de donde estaba antes. Esto puede usarse para explicar y entender un poco mejor el razonamiento de esta parte.

Gráfica resultante al hacerlo con $vo = 40 \frac{m}{s}$ y $\theta o = 36^{o}$



c) Para hacer esto deberemos hallar en le función analitica de y el valor del tiempo en el que la altura vale 0 (además de t=0). Este será el tiempo en el que el objeto vuelva a tocar el piso. Luego sustituimos ese valor de t en la ecuación de la posición horizontal. Derivamos esa función hallada según θ. Luego podremos maximizar la función y ver para qué valor de θ se consigue el alcance máximo, porque en ese punto su derivada valdrá 0.

Tenemos que

$$y(t) = yo + voy * t - g * t^2/2 \Rightarrow 0 = 0 + vo * sen(\theta) * t - g * t^2/2$$

$$vo * sen(\theta) * t - g * \frac{t^2}{2} = 0 \Rightarrow t(vo * sen(\theta) - g * \frac{t}{2}) = 0$$

$t = 0$ (Cuando sale del piso, no es el que buscamos)

$$vo * sen(\theta) - g * \frac{t^*}{2} = 0 \Rightarrow \frac{t^*}{2} = \frac{vo*sen(\theta)}{g} \Rightarrow t^* = \frac{2*vo*sen(\theta)}{g}$$

Sustituimos el tiempo hallado en la ecuación de x(t) para hallar el alcance:

$$x(t) = xo + vox * t \Rightarrow x(\frac{2*vo*sen(\theta)}{g}) = 0 + vo * cos(\theta) * \frac{2*vo*sen(\theta)}{g} \Rightarrow$$

$$xalc = x(\frac{2*vo*sen(\theta)}{g}) = \frac{2*vo^2*sen(\theta)*cos(\theta)}{g}$$

Derivamos la función hallada según θ para maximizar

$$xalc' = x'(t^*) = \frac{\delta x(vo,\theta)}{\delta\theta} = [\frac{2*vo^2}{g} * (sen(\theta) * cos(\theta))]'$$

$$x'(t^*) = (\frac{2*vo^2}{g})' * (sen(\theta) * cos(\theta)) + \frac{2*vo^2}{g} * (sen(\theta) * cos(\theta))'$$

$$x'(t^*) = 0 * (sen(\theta) * cos(\theta)) + \frac{2*vo^2}{g} * (sen(\theta)' * cos(\theta) + sen(\theta) * cos(\theta)')$$

$$x'(t^*) = \frac{2*vo^2}{g} * (cos(\theta) * cos(\theta) - sen(\theta) * sen(\theta))$$

$$x'(t^*) = \frac{2*vo^2}{g} * (cos^2(\theta) - sen^2(\theta))$$

Ahora procedemos a maximizar la derivada

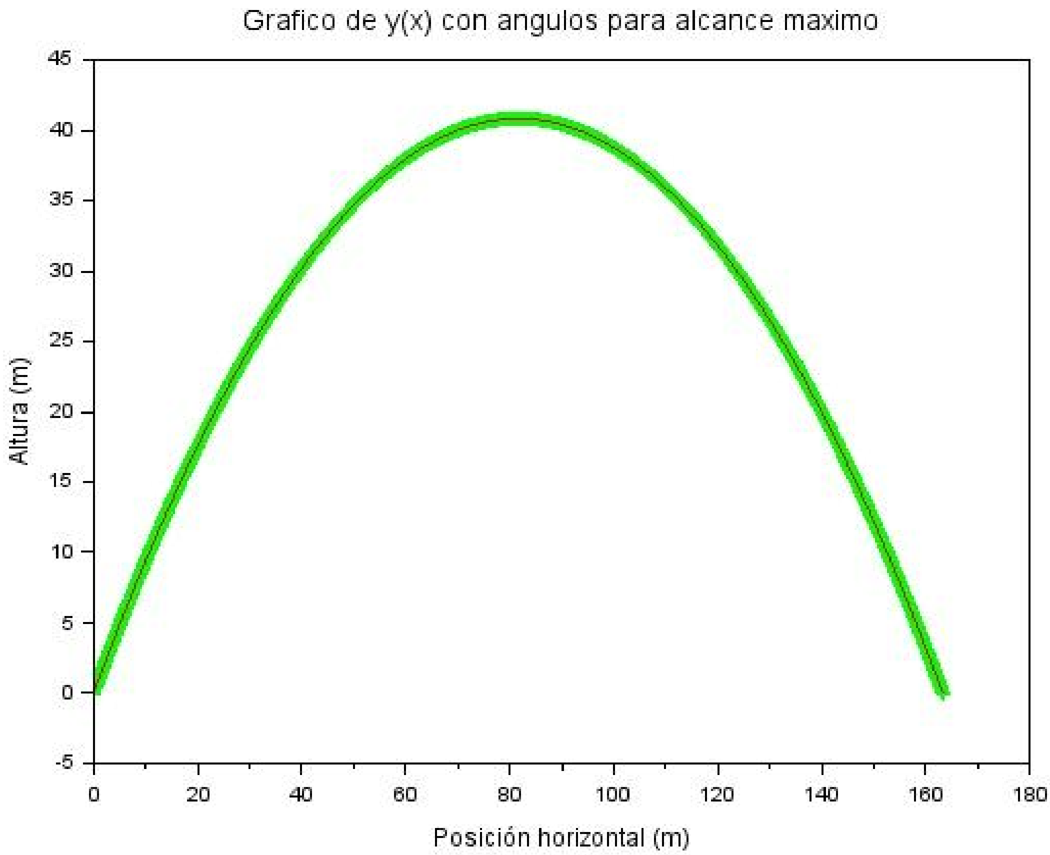
$$x'(t^*) = 0 \Rightarrow \frac{2*vo^2}{g} * (cos^2(\theta) - sen^2(\theta)) = 0$$

como $\frac{2*vo^2}{g} \neq 0 \Rightarrow cos^2(\theta) - sen^2(\theta) = 0$

$$cos^2(\theta) = sen^2(\theta) \Rightarrow cos(\theta) = sen(\theta) \Leftrightarrow \theta = \frac{\pi}{4} = 45^o$$

d) Se implementa un algoritmo que va calculando el alcance del objeto según el ángulo a través de su trayectoria. Se trata de una iteración que va variando el valor del ángulo en una diferencia dada (medio grado en este caso). Cuando se llega al alcance máximo se sale del loop y nos quedamos con el ángulo que se buscaba.

e) Utilizamos los métodos de la parte a y b para calcular y graficar la trayectoria del objeto con los ángulos hallados en las partes anteriores. A continuación, se adjunta la gráfica de $y(x)$ con $vo\ = 40\ \frac{m}{s}$:



__2. Caso de proyectil con influencia del aire:__

a) Primero debemos calcular los componentes de la aceleración en las dimensiones x e y. Sabemos que F se opone a vo y conocemos la ecuación de su módulo, entonces podemos afirmar que $F =- \frac{1}{2}\rho A v \vec{v}$.

Desarrollando esto llegamos a $F =- \frac{1}{2}\rho A\sqrt{vx^2 + vy^2}(vx\hat{x} + vy\hat{y})$. Ya podemos descomponer y hallar los componentes de la fuerza:

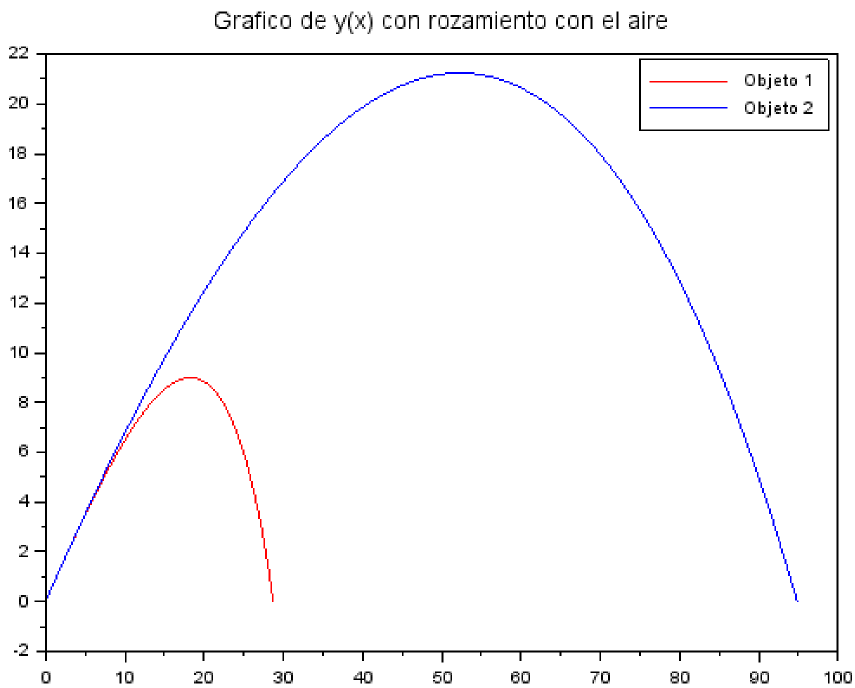
$Fx =- \frac{1}{2}\rho A\sqrt{vx^2 + vy^2}vx\hat{x}$ y $Fy =- \frac{1}{2}\rho A\sqrt{vx^2 + vy^2}vy\hat{x}$

Ahora podemos hallar el valor de la aceleración en cada eje con la segunda ley de Newton:

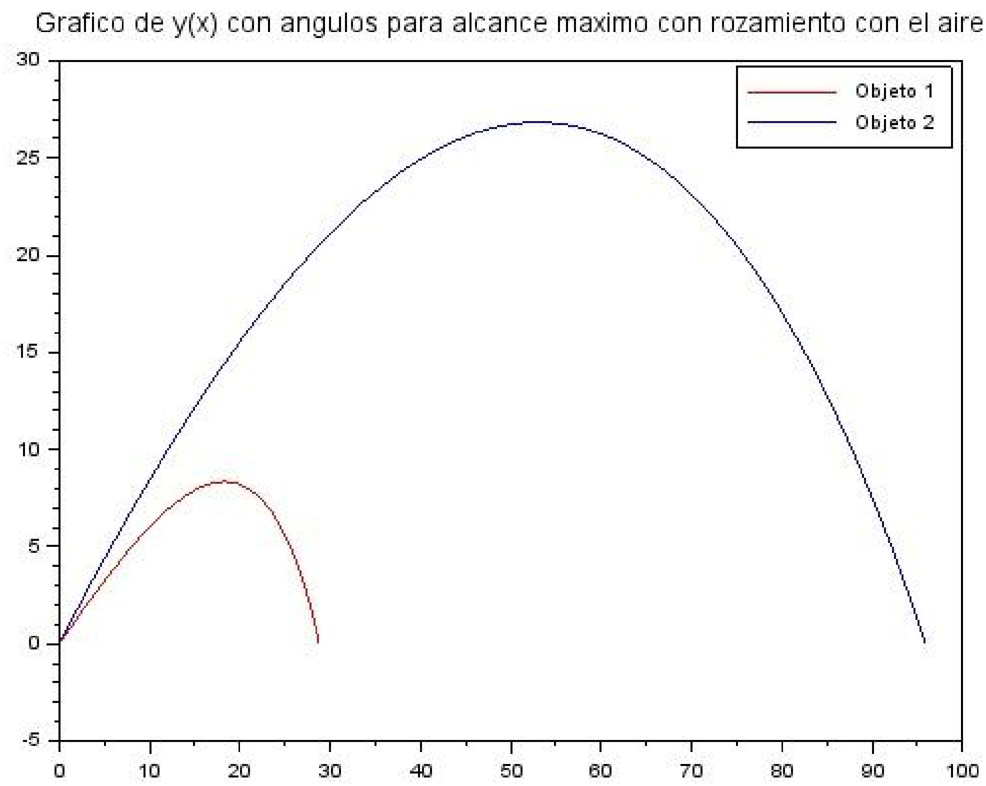
$$Fx = m * ax \Rightarrow ax =- \frac{1}{2}\frac{\rho A}{m}\sqrt{vx^2 + vy^2}vx$$

$$Fy + mg = m * ay \Rightarrow ay =- \frac{1}{2}\frac{\rho A}{m}\sqrt{vx^2 + vy^2}vy - g$$

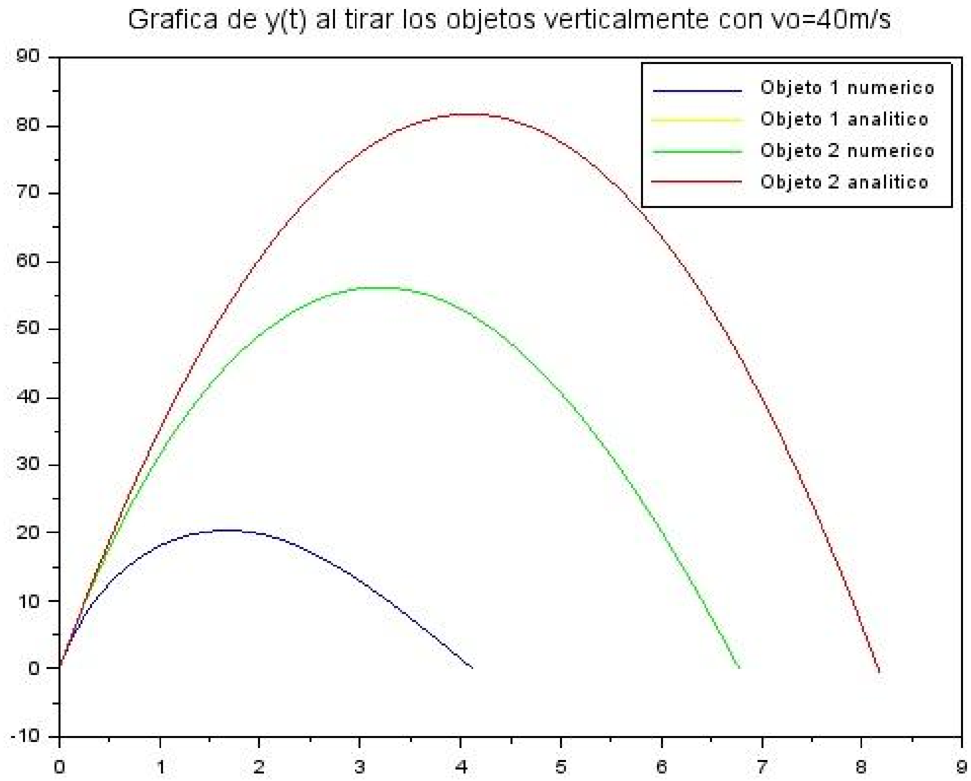
Solo resta aplicar el método de Euler para hallar la trayectoria. El algoritmo en el código será similar al del ejercicio b de la parte anterior. Deberemos cambiar el valor de la aceleración por el que calculamos. Gráfica obtenida al hallar las trayectorias de ambos objetos con este método, tomando $vo = 40\frac{m}{s}$ y $\theta = 36°$



b) Para hallar el ángulo máximo usaremos el concepto general del código usado en la parte 1-d). Básicamente, calcularemos la trayectoria del objeto y vemos su alcance, comparándolo con el conseguido con el valor del ángulo anterior. Esto se va a hacer mientras el alcance anterior sea menor que el actual. Cuando el alcance que se calcula es mayor al que se tenía, sabemos que el ángulo anterior es el máximo.
Luego, solo resta usar el algoritmo que ya habíamos hecho para hallar la trayectoria de un objeto teniendo en cuenta el rozamiento con $vo = 40\frac{m}{s}$ y θ siendo igual al ángulo máximo hallado.
Se adjunta la gráfica:



c) Para este ejercicio, por un lado volveremos a usar el código del ejercicio uno de esta segunda parte. De ahí podremos usar el método de Euler tal como lo definimos, tomando $vo = 40\frac{m}{s}$ y $\theta = 90°$. También deberemos usar el código de la primera consigna para hallar la trayectoria sin rozamiento de manera analitica, con los mismos valores de vo y θ. Algo relevante es que ambos bloques de código que reutilizaremos deberán ser modificados levemente para graficar la posición vertical en función del tiempo, no en función de la distancia horizontal como lo hacían en partes anteriores.



Código de Scilab

```
clear();
//Ingresamos todos los datos
x(1)=0;
y(1)=0;
printf("Ingrese el valor de la velocidad inicial sin rozamiento");
vo = input('');
printf("Ingrese el valor del ángulo inicial sin rozamiento");
tita = input('');
g = 9.8;
vx(1)=vo*cosd(tita); //Genero condiciones iniciales
vy(1)=vo*sind(tita);
ax(1)= 0;
ay(1)= -g;

//PRIMERA PARTE
k = 1;
dt = 0.001;

//Numerico, parte b
while y(k)>=0
   ax(k+1)= ax(k)
   ay(k+1)= ay(k)
   vx(k+1)= vx(k)+ax(k)*dt
   vy(k+1)= vy(k)+ay(k)*dt
   x(k+1)= x(k)+vx(k)*dt
   y(k+1)= y(k)+vy(k)*dt
   k=k+1
end
t=[0:dt:(k-1)*dt]'

//Analitico, parte a
ya=y(1)+vy(1)*t +(ay(1)/2)*t^2 //Hallamos y en funcion del tiempo
xa=x(1)+vx(1)*t //Hallamos x en funcion del tiempo

//Alcance
alcance = x($) //$ es siempre el ultimo valor del vector

//Parte a
```

```
scf(1)
plot(xa,ya,"-r") //Graficamos y en función de x
xlabel('Posición horizontal (m)')
ylabel('Altura (m)')
title('Grafico de y(x) hallado de manera analítica')

//Parte b
scf(2)
plot(x,y,"+g") //Graficamos y en función de x
xlabel('Posición horizontal (m)')
ylabel('Altura (m)')
title('Grafico de y(x) hallado de manera numérica')

//Parte d
titan = 0;
alcanceAnt = 0;
alcanceMax = 0;
while alcanceMax >= alcanceAnt
    titan = titan + 0.5
    alcanceAnt = alcanceMax
    alcanceMax = 2*vo^2*sind(titan)*cosd(titan)/g
end
titan = titan - 0.5 //Vamos a sumar medio grado de mas en el while, se lo
restamos

//Parte e
//GRAFICAR CON LOS ANGULOS DE C Y D
titaa = 45; //Angulo analitico de la parte c
vo = 40;
xn(1)=0;
yn(1)=0;
xa(1)=0;
ya(1)=0;
axn(1)=0;
ayn(1)=-g;
//Numerica
vxn(1)=vo*cosd(titan);
vyn(1)=vo*sind(titan);
k=1;
while yn(k)>=0
```

```
    axn(k+1)= axn(k)
    ayn(k+1)= ayn(k)
    vxn(k+1)= vxn(k)+axn(k)*dt
    vyn(k+1)= vyn(k)+ayn(k)*dt
    xn(k+1)= xn(k)+vxn(k)*dt
    yn(k+1)= yn(k)+vyn(k)*dt
    k=k+1
end
t=[0:dt:(k-1)*dt]'
scf(3)
plot(xn,yn,"+g")
xlabel('Posición horizontal (m)')
ylabel('Altura (m)')
title('Grafico de y(x) con angulos para alcance maximo')

//Analitica
vxa(1)=vo*cosd(titaa);
vya(1)=vo*sind(titaa);
yaa=y(1)+vya(1)*t +(ay(1)/2)*t^2 //Hallamos y en funcion del tiempo
xaa=xa(1)+vxa(1)*t
plot(xaa,yaa,"-r")

//SEGUNDA PARTE
//Parte a
x1(1)=0;
y1(1)=0;
x2(1)=0;
y2(1)=0;
printf("Ingrese el valor de la velocidad inicial con rozamiento");
vo = input('');
printf("Ingrese el valor del ángulo inicial con rozamiento");
tita = input('');
vx1(1)=vo*cosd(tita); //Genero condiciones iniciales
vy1(1)=vo*sind(tita);
vx2(1)=vo*cosd(tita); //Genero condiciones iniciales
vy2(1)=vo*sind(tita);
g=9.8;
p = 1.2;

m1=0.4;
```

```
r1=0.11; //Lo pasamos a metros
area1=%pi*r1^2; //Calculamos area
ax1(1)=-(1/2)*(p*area1/m1)*sqrt(vx1(1)^2+vy1(1)^2)*vx1(1)
ay1(1)=-(1/2)*(p*area1/m1)*sqrt(vx1(1)^2+vy1(1)^2)*vy1(1)-g
//Graficamos
k = 1;
dt = 0.001;
while y1(k)>=0
    ax1(k+1)= -(1/2)*(p*area1/m1)*sqrt(vx1(k)^2+vy1(k)^2)*vx1(k)
    ay1(k+1)= -(1/2)*(p*area1/m1)*sqrt(vx1(k)^2+vy1(k)^2)*vy1(k)-g
    vx1(k+1)= vx1(k)+ax1(k)*dt
    vy1(k+1)= vy1(k)+ay1(k)*dt
    x1(k+1)= x1(k)+vx1(k)*dt
    y1(k+1)= y1(k)+vy1(k)*dt
    k=k+1
end
t=[0:dt:(k-1)*dt]'
scf(4)
plot(x1,y1,-1,"color","red")

m2=0.03;
r2=0.01; //Lo pasamos a metros
area2=%pi*r2^2;
ax2(1)=-(1/2)*(p*area2/m2)*sqrt(vx2(1)^2+vy2(1)^2)*vx2(1)
ay2(1)=-(1/2)*(p*area2/m2)*sqrt(vx2(1)^2+vy2(1)^2)*vy2(1)-g
//Graficamos
k = 1;
dt = 0.001;
while y2(k)>=0
    ax2(k+1)= -(1/2)*(p*area2/m2)*sqrt(vx2(k)^2+vy2(k)^2)*vx2(k)
    ay2(k+1)= -(1/2)*(p*area2/m2)*sqrt(vx2(k)^2+vy2(k)^2)*vy2(k)-g
    vx2(k+1)= vx2(k)+ax2(k)*dt
    vy2(k+1)= vy2(k)+ay2(k)*dt
    x2(k+1)= x2(k)+vx2(k)*dt
    y2(k+1)= y2(k)+vy2(k)*dt
    k=k+1
end
t=[0:dt:(k-1)*dt]'
scf(4)
plot(x2,y2,-1)
```

```
title('Grafico de y(x) con rozamiento con el aire')
legend(["Objeto 1","Objeto 2"])

//Parte b

//Con el objeto 1, la pelota
clear(['x1' 'x2' 'y2' 'y1'])
titan1 = 0;
m1=0.4;
r1=0.11;
area1=%pi*r1^2;
p = 1.2;
vo = 40
g=9.8;
alcanceAnt1 = 0;
alcanceMax1 = 0;
b = 0
while alcanceMax1 >= alcanceAnt1
    titan1 = titan1 + 1
    vx1(1)=vo*cosd(titan1);
    vy1(1)=vo*sind(titan1);
    x1(1) = 0
    y1(1) = 0
    ax1(1)=-(1/2)*(p*area1/m1)*sqrt(vx1(1)^2+vy1(1)^2)*vx1(1)
    ay1(1)=-(1/2)*(p*area1/m1)*sqrt(vx1(1)^2+vy1(1)^2)*vy1(1)-g
    k = 1
    alcanceAnt1 = alcanceMax1
    dt = 0.001;
    while y1(k)>=0
        ax1(k+1)= -(1/2)*(p*area1/m1)*sqrt(vx1(k)^2+vy1(k)^2)*vx1(k)
        ay1(k+1)= -(1/2)*(p*area1/m1)*sqrt(vx1(k)^2+vy1(k)^2)*vy1(k)-g
        vx1(k+1)= vx1(k)+ax1(k)*dt
        vy1(k+1)= vy1(k)+ay1(k)*dt
        x1(k+1)= x1(k)+vx1(k)*dt
        y1(k+1)= y1(k)+vy1(k)*dt
        k=k+1
    end
    t=[0:dt:(k-1)*dt]'
    alcanceMax1 = x1($)
end
```

```
t=[0:dt:(k-1)*dt]'
scf(5)
plot(x1,y1,+1,"color","red")
title('Grafico de y(x) con angulos para alcance maximo con rozamiento con el
aire')

titan2 = 0;
m2=0.03;
r2=0.01;
area2=%pi*r2^2;
p = 1.2;
vo = 40
g=9.8;
alcanceAnt2 = 0;
alcanceMax2 = 0;
b = 0
while alcanceMax2 >= alcanceAnt2
    titan2 = titan2 + 1
    vx2(1)=vo*cosd(titan2);
    vy2(1)=vo*sind(titan2);
    x2(1) = 0
    y2(1) = 0
    ax2(1)=-(1/2)*(p*area2/m2)*sqrt(vx2(1)^2+vy2(1)^2)*vx2(1)
    ay2(1)=-(1/2)*(p*area2/m2)*sqrt(vx2(1)^2+vy2(1)^2)*vy2(1)-g
    k = 1
    alcanceAnt2 = alcanceMax2
    dt = 0.001;
    while y2(k)>=0
        ax2(k+1)= -(1/2)*(p*area2/m2)*sqrt(vx2(k)^2+vy2(k)^2)*vx2(k)
        ay2(k+1)= -(1/2)*(p*area2/m2)*sqrt(vx2(k)^2+vy2(k)^2)*vy2(k)-g
        vx2(k+1)= vx2(k)+ax2(k)*dt
        vy2(k+1)= vy2(k)+ay2(k)*dt
        x2(k+1)= x2(k)+vx2(k)*dt
        y2(k+1)= y2(k)+vy2(k)*dt
        k=k+1
    end
    t=[0:dt:(k-1)*dt]'
    alcanceMax2 = x2($)
end
t=[0:dt:(k-1)*dt]'
```

```
plot(x2,y2,+1)
legend(["Objeto 1","Objeto 2"])

//Parte c

//Objeto 1 numerico
x1(1)=0;
y1(1)=0;
vo = 40;
tita = 90;
vx1(1)=vo*cosd(tita);
vy1(1)=vo*sind(tita);
g=9.8;
p = 1.2;
m1=0.4;
r1=0.11;
area1=%pi*r1^2
ax1(1)=-(1/2)*(p*area1/m1)*sqrt(vx1(1)^2+vy1(1)^2)*vx1(1)
ay1(1)=-(1/2)*(p*area1/m1)*sqrt(vx1(1)^2+vy1(1)^2)*vy1(1)-g
k = 1;
dt = 0.001;
altMax1 = 0
tiempo1 = 0
while y1(k)>=0
    ax1(k+1)= -(1/2)*(p*area1/m1)*sqrt(vx1(k)^2+vy1(k)^2)*vx1(k)
    ay1(k+1)= -(1/2)*(p*area1/m1)*sqrt(vx1(k)^2+vy1(k)^2)*vy1(k)-g
    vx1(k+1)= vx1(k)+ax1(k)*dt
    vy1(k+1)= vy1(k)+ay1(k)*dt
    x1(k+1)= x1(k)+vx1(k)*dt
    y1(k+1)= y1(k)+vy1(k)*dt
    tiempo1 = tiempo1 + 0.001
    if y1(k+1)<y1(k) && altMax1==0
        altMax1 = y1(k)
        tiempoMax1 = tiempo1
    end
    k=k+1
end
t=[0:dt:(k-1)*dt]'
scf(6)
plot(t,y1,-1)
```

```
title("Grafica de y(t) al tirar los objetos verticalmente con vo=40m/s")
//Objeto 1 analitico
vx1(1)=0
vy1(1)=40*sind(90)
ay1(1)=-g
x1(1)=0
y1(1)=0
t2=[0:dt:(k+1400)*dt]'
ya1=y1(1)+vy1(1)*t2 +(ay1(1)/2)*t2^2
xa1=x1(1)+vx1(1)*t2
scf(6)
plot(t2,ya1,-1,"color","yellow")

//Objeto 2 numerico
x2(1)=0;
y2(1)=0;
vo = 40;
tita = 90;
vx2(1)=vo*cosd(tita);
vy2(1)=vo*sind(tita);
g=9.8;
p = 1.2;
m2=0.03;
r2=0.01;
area2=%pi*r2^2
ax2(1)=-(1/2)*(p*area2/m2)*sqrt(vx2(1)^2+vy2(1)^2)*vx2(1)
ay2(1)=-(1/2)*(p*area2/m2)*sqrt(vx2(1)^2+vy2(1)^2)*vy2(1)-g
k = 1;
dt = 0.001;
altMax2 = 0
tiempo2 = 0
while y2(k)>=0
   ax2(k+1)= -(1/2)*(p*area2/m2)*sqrt(vx2(k)^2+vy2(k)^2)*vx2(k)
   ay2(k+1)= -(1/2)*(p*area2/m2)*sqrt(vx2(k)^2+vy2(k)^2)*vy2(k)-g
   vx2(k+1)= vx2(k)+ax2(k)*dt
   vy2(k+1)= vy2(k)+ay2(k)*dt
   x2(k+1)= x2(k)+vx2(k)*dt
   y2(k+1)= y2(k)+vy2(k)*dt
   tiempo2 = tiempo2 + 0.001
   if y2(k+1)<y2(k) && altMax2==0
```

```
        altMax2 = y2(k)
        tiempoMax2 = tiempo2
    end
    k=k+1
end
t=[0:dt:(k-1)*dt]'
scf(6)
plot(t,y2,-1,"color","green")
//Objeto 2 analitico
vx2(1)=0
vy2(1)=40*sind(90)
ay2(1)=-g
x2(1)=0
y2(1)=0
t2=[0:dt:(k+1400)*dt]'
ya2=y2(1)+vy2(1)*t2 +(ay2(1)/2)*t2^2
xa2=x2(1)+vx2(1)*t2
plot(t2,ya2,-1,"color","red")
legend(["Objeto 1 numerico","Objeto 1 analitico","Objeto 2 numerico","Objeto 2 analitico"])
```